פרויקט חלק ב'

חלק 1 - ETL

PART A – Build table 'StreamingWeather'

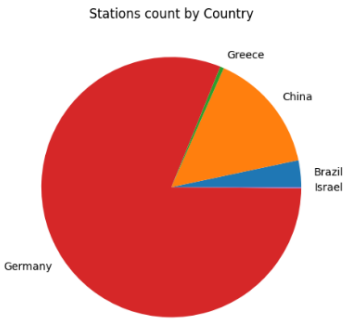
Attribute name	type	explanation
Batch_id	VARCHAR	מזהה batch
StationId		מזהה תחנה
FIPS_code		קוד מדינה
Month	int	חודש
Year		שנה
SumTmax	float	סכום של 'x' variable לכל תחנה, בחודש ושנה **עשוי להכיל Nulls
SumTmin		
SumPrcp		
countTmax	int	מספר רשומות שונות מnull של 'x' variable לכל תחנה, בחודש ושנה
countTmin		
countPrcp		
LATITUDE	float	קווי גובה
LONGITUDE		קווי רוחב
ELEVATION		גובה מעל פני קרקע

בחירת משתנים :

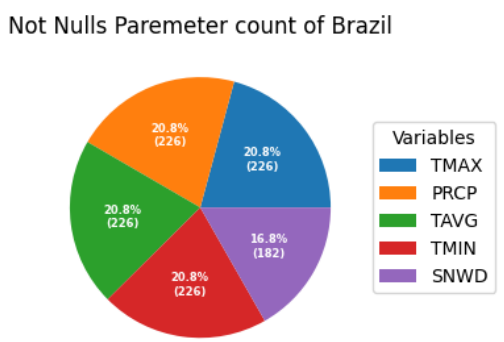
1. יצאנו מנקודת הנחה כי הפרמטרים הבאים יעזרו לנו לחזות בצורה הטובה ביותר את המשקעים, זאת לאור הקשר ההדוק בין משקעים ומזג האוויר: 'TMAX', 'TMIN', 'PRCP'.

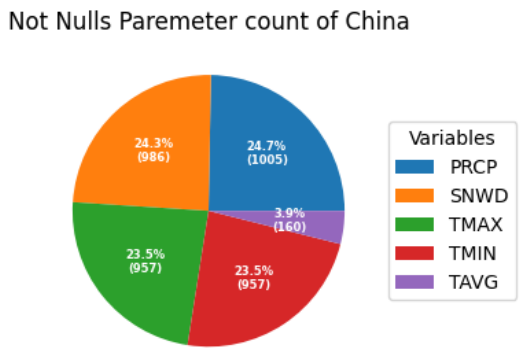
טרם בחירת המשתנים לעיל הרצנו אנליזות על הנתונים בכדי לאתר את המשתנים אשר מכילים כמה שפחות ערכי nulls. בחרנו לבצע את האנליזה על הטבלה inventory על מנת לאתר את המשתנים אשר lastyear שלהם הייתה לאחר שנת 1970 או לחילופין firstyear הייתה אחרי 1970:

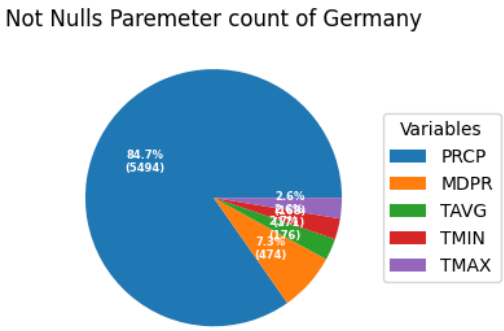
להלן התפלגות כמות התחנות לפי מדינה:

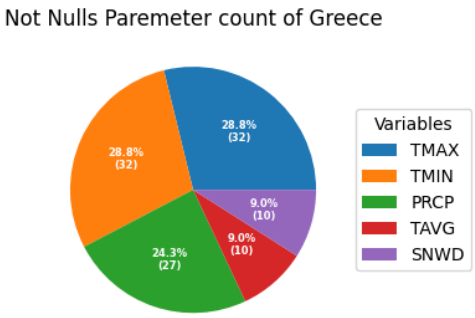


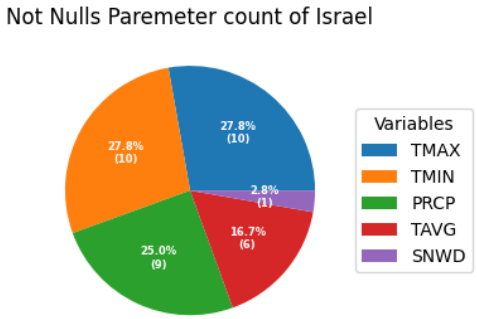
להלן כמות ערכי ה-variables אשר אינם null פר מדינה:











מאנליזות אלו הסקנו כי המשתנים הנפוצים ביותר בקרב כל התחנות שגם קיימים בכמות דומיננטית יחסית

הם: PRCP, TMAX, TMIN, TAVG- בחרנו לוותר על TAVG כי הוא מופיע מעט יותר וכי טמפ' מנימלית ומקסימלית הספיקו לצרכינו. בחרנו בפרמטרים שמכילים כמה שפחות ערכי null על מנת שיהיו לנו כמה שיותר רשומות משמעותיות בטבלה בשביל ביצוע שאר שלבי הפרויקט באופן מיטבי.

2. הוספנו עמודות של מיקום על פני כדור הארץ מהטבלה ghcnd-stations כדי שנוכל להעריך את הקשר בין המיקום והגובה מעל פני הקרקע לבין כמות המשקעים.

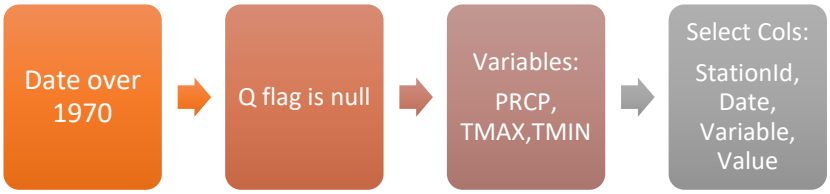
<u>בחירת מדינות:</u>

ישראל, סין, גרמניה, יוון, ברזיל.

בחרנו את ישראל ויוון מאחר והאקלים בשתיהן יחסית דומה. מצד שני גרמניה גשומה מאד ורחוקה יחסית מהשתיים הראשונות. סין וברזיל נמצאות בקצוות של כדור הארץ. האקלים בברזיל הוא אקלים טרופי אשר שונה מכל שאר המדינות. סין מדינה עצומה עם הרבה מאד תחנות.

המדינות שבחרנו בחלקן קרובות אחת לשנייה, אבל באופן כללי שונות אחת מהשנייה, הן ביבשת והן בגודל ובסוג האקלים (טרופי לעומת ים תיכוני למשל).

סינון ראשוני:



שנים : פילטרנו על רשומות עדכניות יחסית מ-1970 ואילך. זאת מאחר ורצינו להסתמך בניתוחים שלנו על נתונים יחסית עדכניים אשר מרבית הסיכויים שהם יותר אמינים (לאור התפתחות הטכנולוגיה והמחשב במאה ה20).

אמינות: פילטרנו על נתונים אשר הכילו null ב-Q_Flag . לפי הקובץ של הפרויקט רשומות אלו לא נכשלו בבדיקת איכות ולכן הסקנו כי האמינות שלהן היא הגבוהה ביותר.

משתנים: השארנו משתנים רלוונטיים לפי ההצדקות שפירטנו קודם.

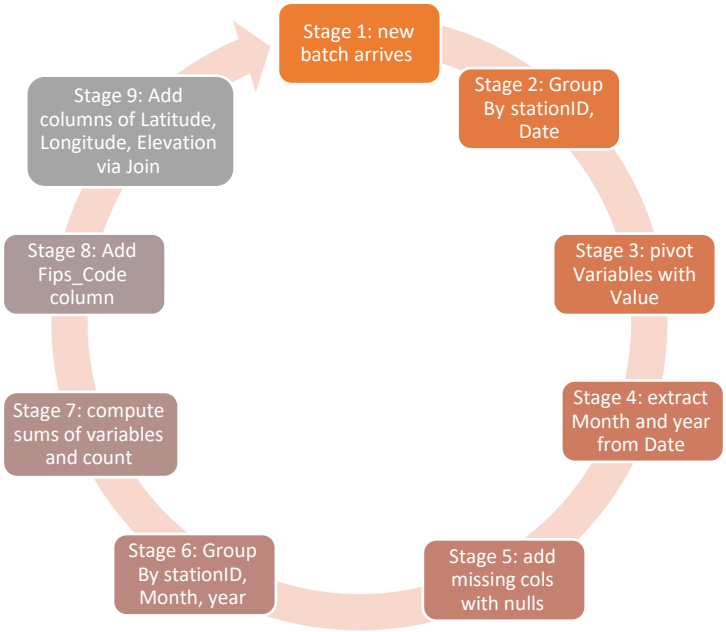
עמודות: בחרנו להתעלם מהעמודות של Flags, מאחר ולא ראינו בהן ערך מוסף.

Part B - Streaming

הקדמה: בעת הזרמת הנתונים מ-kafka אנחנו מעבדים את ה-batches עם פונקציית forachBatch בעזרתה אנחנו מבצעים טרנספורמציות שונות (יוסברו בהמשך) על הנתונים ומזרימים את הנתונים המעובדים לשרת sql לטבלה העזר streamingWeather. הגדרנו כי כל batch יכיל לכל היותר 500,000 רשומות.

המטרה : לשמור לכל תחנה, בחודש ובשנה את הממוצע של כל אחד מהמדדים. מאחר ואנחנו קולטים את הנתונים ב-streaming אין לנו אפשרות לבנות את הממוצע לכל batch מאחר ויכולות להיות רשומות רלוונטיות לתחנה בחודש ובשנה ב-batch שונים. לכן החלטנו ללכת בגישה אשר שומרת את ה- stateful transformation בטבלה הראשית לאחר הטרנספורמציות המרכזיות, ובסוף ההזרמה נבצע את הממוצע על ידי טרנספורמציות על spark df.

טרנספורמציות של foreachfunc:



להלן פירוט של השלבים המתאימים בתרשים:

1. ספרנו את מס' ה-batch-ים שקלטנו על ידי משתנה גלובלי, כדי שנוכל לעקוב אחרי כמות הרשומות שאנחנו קולטים מה-stream.
2. ביצענו אגרגציה לפי תאריך ומזהה תחנה, זאת כדי לאגד את כל המשתנים שרלוונטיים לאותו יום ולהתאים את הערכים של המדידה בתאריך ובתחנה.
3. בעזרת Pivot הפכנו את המשתנים שבחרנו מתוך העמודה variable לעמודות בשמות מתאימים.
4. חילצנו מהתאריך את החודש והשנה אשר ישמש לשם אגרציה בסעיפים הבאים.
5. במידה ולא נקלטה דגימה של אחד הפרמטרים, נרצה לתעד זאת ולכן נוסיף עמודה של null שתאפשר הכנסה תקינה של הBatch למסד.

Example:

Batch x has the following rows (as dataframe):

stationID	Date	Variable	Value
CH1234	01021997	PRCP	20
CH1234	01021997	TMAX	30
CH1235	02021997	PRCP	45
CH1235	02021997	TMAX	35

Batch after pivot is:

stationID	Date	PRCP	TMAX
CH1234	01021997	20	30
CH1234	01021997	20	30
CH1235	02021997	45	35
CH1235	02021997	45	35

Batch after adding nulls columns:

stationID	Date	PRCP	TMAX	TMIN
CH1234	01021997	20	30	null
CH1234	01021997	20	30	null
CH1235	02021997	45	35	null
CH1235	02021997	45	35	null

6. ביצענו אגרציה מחדש לפי חודש, שנה ותחנה.
7. הוספנו עמודות שסוכמות על פני כל מדד, וכן עמודות שסופרות את כמות הרשומות המתאימות (ישמש בהמשך לחישוב הממוצע האגרגטיבי על פני כל הbatches).
8. הוספנו עמודה של FIPS_CODE שמייצג את המדינה, אשר חילצנו מתוך מזהה התחנה. נשתמש בפרמטר זה בהמשך בשלב של ניתוח הנתונים.
9. הוספנו עמודות שמכילות מידע על המיקום (קווי אורך ורוחב, וגובה מעל פני הקרקע) של כל תחנה, על ידי join עם dataframe שיצרנו מתוך הקובץ ghcnd-stations.
10. הוספנו batchID אשר מאפשר לנו גמישות בעמודה עם הרלציה הראשית. אם לא היינו עושים זאת היינו נתקלים בבעיה של מפתח ראשי, זאת מאחר ויכול להיות מצב בו רשומות ששייכות לאותה תחנה, חודש ושנה, מגיעות ל-batch-ים שונים.

בסוף נכתוב את הדאטא החדש (שעבר את שלבים 1-10) לטבלה הראשית.

PART C- Apply average into final table 'Weather'

לאחר שקלטנו את הנתונים ועיבדנו אותם, אנחנו מוכנים לבצע איגוד אחרון, ולחשב את הממוצע של כל אחד מהמדדים לחודש, שנה ותחנה.

למשל עבור Tmin: avg = sum(sumTmin)/sum(countTmin).

Attribute name	type	explanation
StationId		מזהה תחנה
FIPS_code		קוד מדינה

Month	int	חודש
Year	int	שנה
AvgTmax	float	ממוצע של variable 'x' לכל תחנה, בחודש ושנה
AvgTmin		
AvgPrcp		
LATITUDE	float	קווי גובה
LONGITUDE		קווי רוחב
ELEVATION		גובה מעל פני קרקע

ביצענו זאת באמצעות spark df בתוך פונקציה -finishETL. פונקציה זו מייצרת spark df מהטבלה streamingWeather מבצעת את האגרגציות ומוחקת רשומות המכילות ערכי null.
לבסוף נייצא את הטבלה הגמורה לטבלת Weather.